Arab J Sci Eng (2012) 37:697–707
DOI 10.1007/s13369-012-0196-3

RESEARCH ARTICLE - ELECTRICAL ENGINEERING

Mohammad Javad Maghsoudi · Zuwairie Ibrahim · Salinda Buyamin · Mohd Fua'ad Rahmat

Data Clustering for the DNA Computing Readout Method Implemented on LightCycler and Based on Particle Swarm Optimization

Received: 1 March 2010 / Accepted: 7 October 2010 / Published online: 28 February 2012

Abstract In this work, particle swarm optimization (PSO) is applied to automate the DNA computing readout method based on a real-time polymerase chain reaction (PCR). Moreover, real-time amplification was performed and the TaqMan detection approach was used for the plan and the readout approach development. The most important part of the readout method is identifying two different reactions in the real-time PCR, which involve in vitro and in silico processes in order to inspect the placement of pairs of nodes in the Hamiltonian path problem. In addition, the real-time PCR experiment is implemented on the LightCycler System. Previously, manual method was exploited to classify two different output reactions of real-time PCR that was a time consuming process. In this study, by exploiting MATLAB the PSO has been implemented for clustering output reactions of real-time PCR and experimental results depict that the amplification response for "YES" and "NO" reactions can be clustered correctly.

Keywords Particle swarm optimization · DNA computing · Data clustering · Real-time PCR

الخلاصة

تم ـ في هذا العمل ـ تطبيق سرب الجسيمات الامثل (PSO) لأتمتة طريقة حوسبة قراءة الحمض النووي المعتمدة على الوقت الحقيقي لتفاعل سلسلة البوليميريز. إن أهم جزء في طريقة القراءة هو التعرف على اثنين من التفاعلات المختلفة في الوقت الحقيقي PCR الذي يتضمن عمليات في المختبر وفي السيليكو من أجل تفقد وضع أزواج من العقد في مسار مشكلة هاميلتونين (HPP). بالإضافة إلى ذلك تم تطبيق تجربة الوقت الحقيقي PCR على نظام LightCycler و MATLAB لتنفيذ تطبيق سرب الجسيمات الأمثل للتجمع.

1 Introduction

Based on the findings of the polymerase chain reaction (PCR) [1], the following innovation in real-time is commonly employed and has played a significant role in molecular medicine and clinical diagnostics [2]. All real-time amplification instruments require a fluorescence reporter molecule for detection and quantization that the increment in its signal is proportional to the amount of amplified product. A TaqMan DNA probe is a modified nonexpendable dual labeled oligonucleotide. The $5'$ and $3'$ ends of the oligonucleotide are terminated with an attached reporter, such as FAM, and quencher fluorophores dyes, such as TAMRA, respectively, as

M. J. Maghsoudi (✉) · Z. Ibrahim · S. Buyamin · M. F. Rahmat
Faculty of Electrical Engineering, University Technology Malaysia, UTM Skudai, 81310 Johor Bahru, Malaysia
E-mail: maghsoudiab@fkegraduate.utm.my

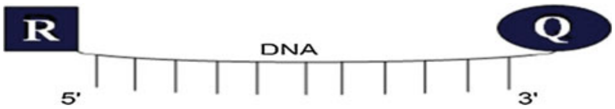

Fig. 1 Illustration of the structure of a TaqMan DNA probe. Here, R and Q denote the reporter and quencher fluorophores, respectively

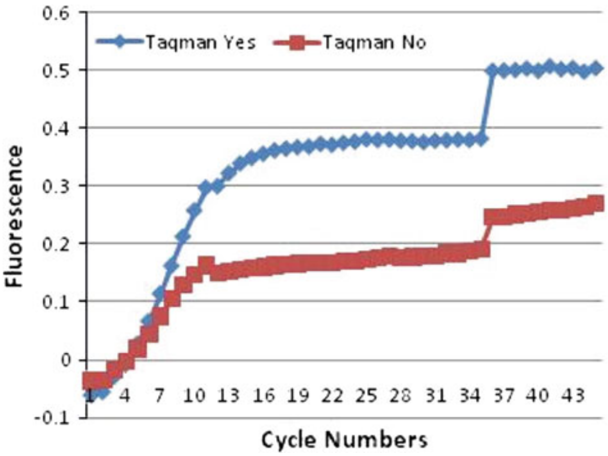

Fig. 2 An instance of reaction plots related to TaqMan (V_0, V_k, V_l) = YES and TaqMan (V_0, V_k, V_l) = NO

shown in Fig. 1 [3]. Upon laser excitation at 488 nm, the FAM fluorophore emits fluorescence at 518 nm in isolation. Given the proximity of the TAMRA quencher, however, and based on the principle of fluorescence resonance energy transfer (FRET), the excitation energy is not emitted by the FAM fluorophore, but rather is transferred to TAMRA via the dipole–dipole interaction between FAM and TAMRA. As TAMRA emits this absorbed energy at significant wavelengths (580 nm), the resulting fluorescence is not observable in Channel 1 of the real-time PCR instruments [4].

Previously, we proposed a readout method tailored specifically to the Hamiltonian path problem (HPP) in DNA computing, which employs a hybrid in vitro/in silico approach [5]. In the in vitro phase, $O(|V|^2)$ TaqMan-based real-time PCR reactions are performed in parallel to investigate the ordering of pairs of nodes in the Hamiltonian path of a $|V|$-node instance graph, in terms of relative distance from the DNA sequence encoding the known start node. The resulting relative orderings are then processed in silico, a process that efficiently returns the complete Hamiltonian path.

The proposed approach is an experimentally validated optical method specifically designed for the quick readout of HPP instances in DNA computing. Previously, graduated PCR, originally demonstrated by Adleman [6], was employed to perform such operations. While a DNA chip based methodology, which makes use of biochip hybridization for the same purpose has been proposed [7,8], this method is more costly and has yet to be experimentally implemented. Previously, Saaid et al. [9] exploited a Fuzzy C-means algorithm, implemented on the LightCycler System, to cluster the output of DNA computing automatically. In this paper, we develop an automated system for the DNA computing readout method implemented on real-time PCR and involving an in vitro/in silico approach. The real-time amplification is performed using TaqMan probes with the TaqMan detection mechanism exploited in the design and development of the readout approach. The in vitro part is performed on the LightCycler System. As shown in Fig. 2, the output of the DNA computing readout method implemented on the LightCycler System consists of two kinds of reactions, namely, “YES” and “NO” reactions. In the in silico information processing, a particle swarm optimization (PSO) clustering algorithm [10] is implemented for automatic classification of the “YES” and “NO” reactions.

2 Readout Approach for DNA Computing Based on Real-Time PCR

2.1 Notation and Basic Principles

In the subsequent text, $V_{1(a)}V_{2(b)}V_{3(c)}V_{4(d)}$ denotes a double stranded DNA (dsDNA), containing the base-pair subsequences, V_1, V_2, V_3, and V_4. Here, the subscripts in parentheses (a, b, c, and d) indicate the

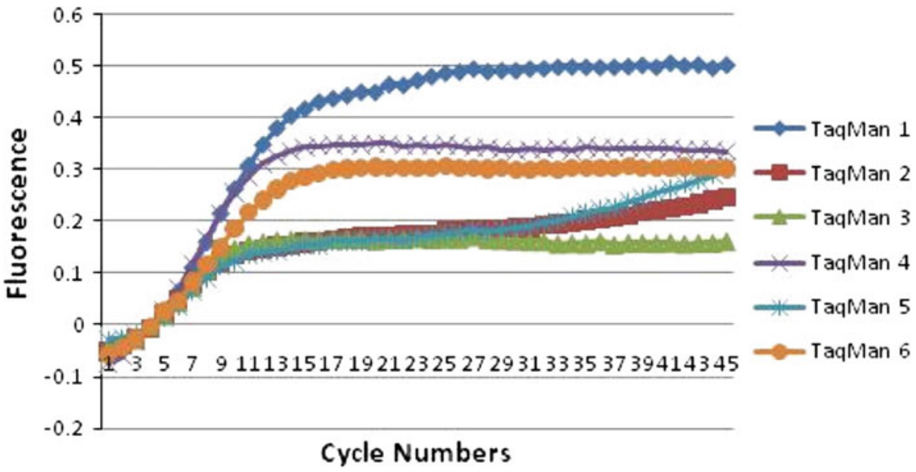

Fig. 3 Output of real-time PCR, reactions 1–6 indicate the $[(|V|-2)2-(|V|-2)]/2$ TaqMan (V_0, V_k, V_l) reactions of the input example

length of each respective base-pair subsequence. For instance, $V_{1(20)}$ indicates that the length of the double-stranded subsequence V_1 is 20 base-pairs (bp). When convenient, a dsDNA may also be represented without indicating the segment lengths (e.g. $V_1V_2V_3V_4$). A reaction denoted by TaqMan (V_0, V_k, V_l) indicates that real-time PCR is performed using the forward primer V_0, reverse primer V_l, and TaqMan probe V_k. In accordance with the proposed method, there are two possible reaction conditions related to the relative positions of the TaqMan probe and reverse primer. In particular, the first condition occurs when the TaqMan probe specifically hybridizes to the template, between the forward and reverse primers, while the second occurs when the reverse primer hybridizes between the forward primer and the TaqMan probe. As shown in Fig. 3, these two conditions result in different amplification patterns during real-time PCR, given the same DNA template (i.e., assuming that they occurred separately, in two different PCR reactions). The higher fluorescent output of the first condition is a typical amplification plot for real-time PCR. In contrast, the low fluorescent output of the second condition reflects the cleavage of a few of the TaqMan probes via DNA polymerase due to the ‘unfavorable’ hybridization position of the reverse primer. Thus, TaqMan (V_0, V_k, V_l) = YES if an amplification plot similar to the first condition is observed, while TaqMan (V_0, V_k, V_l) = NO, if an amplification plot similar to the second condition is observed.

2.1.1 In Vitro Step

Let the output of an in vitro computation of an HPP instance of the input graph be represented by a 120-bp dsDNA $v_{0(20)}v_{2(20)}v_{4(20)}v_{1(20)}v_{3(20)}v_{5(20)}$, where the Hamiltonian path $V_0 \rightarrow V_2 \rightarrow V_4 \rightarrow V_1 \rightarrow V_3 \rightarrow V_5$, begins at node node V_0, ends at node V_5, and contains intermediate nodes V_2, V_4, V_1, and V_3. Note that in practice, only the identities of the start and end nodes, and the presence of all intermediate nodes will be known in advance to characterize the solution path. The specific order of the intermediate nodes within such a path is unknown. The first part of the approach, which is performed in vitro, comprises $[(|V|-2)^2-(|V|-2)]/2$ real-time PCR reactions, each denoted by TaqMan (V_0, V_k, V_l) for all k and l, such that $0 < k < |V|-2, 1 < l < |V|-1$, and $k < l$. In this example instance, so that the DNA template is the dsDNA $v_0v_2v_4v_1v_3v_5$, these six reactions are given as follows:

1. TaqMan $(V_0, V_1, V_2) =$ NO
2. TaqMan $(V_0, V_1, V_3) =$ YES
3. TaqMan $(V_0, V_1, V_4) =$ NO
4. TaqMan $(V_0, V_2, V_3) =$ YES
5. TaqMan $(V_0, V_2, V_4) =$ YES
6. TaqMan $(V_0, V_3, V_4) =$ NO

Note that the overall process consists of a set of parallel real-time PCR reactions, and thus requires $O(1)$ laboratory steps for in vitro amplification. The accompanying space complexity, in terms of the required number of tubes is $O(|V|^2)$. Clearly, only one forward primer is required for all real-time PCR reactions, while the number of reverse primers and TaqMan probes required with respect to the size of input graph is |V| − 3 in both cases. After all real-time PCR reactions have been completed, the in vitro output is subjected to an algorithm for in silico information processing, producing the satisfying Hamiltonian path of the HPP instance in $O(n^2)$ time (here, n denotes the number of vertices).

2.2 In Silico Part

The next step is to use all the information from the six TaqMan reactions to allocate each node of the Hamiltonian path. This can be done by applying the in silico algorithm as follows:

```
 Input: N[0…|V|-1]=2 // N[0, ?, ?, ?, ?, 5]
 A[1…|V|-2]=|V| // A[1, 1, 1, 1]
  for k=1 to |V|-3
    for l=k+1 to |V|-2
      if TaqMan($V_0$, $V_k$, $V_l$) = YES
         A[l] = A[l]+1
        else A[k] = A[k]+1
       endif
     endfor
   N[ A[k] ] = k
 endfor
N[ A[ |V|-2] ]=|V|-2
```

In this algorithm, an array (N[0...|V| − 1]) is defined to store all the nodes of the Hamiltonian path. In addition, an array of aggregation values (A[1 ... |V| − 2]), used to locate the Hamiltonian path in each array of nodes, is also defined. Based on the modified algorithm, the input array N is first initialized to N = {0, ?, ?, ?, ?, 5} since the start and end nodes of the path are known in advance. Next, the aggregation array A is initialized to A = {1, 1, 1, 1}. During the loop operations of the algorithm, the values in array A are increased in each iteration step. The aggregation array A[i] is used to index the node array for each value of k. After the loop operation, |V| − 2 is assigned to cell N[A[|V| − 2]. The output of the in silico algorithm can be viewed by calling back all the cells in the node array N[0] to N[|V| − 1]. The outcome of the in silico algorithm is N = {0, 2, 4, 1, 3, 5}. Note that this algorithm can be carried out once all the information for the TaqMan reactions has been obtained. This can only be done if clustering is applied to investigate the "YES" and "NO" reactions.

3 Particle Swarm Optimization

Particle swarm optimization (PSO) is a population-based stochastic search process, modeled on the social behavior of a flock of birds [11,12]. In PSO, the assumed birds, called particles, fly in a virtual space to find the optimum of a predefined fitness function. PSO has been exploited across a vast area of research [13]. Any optimization problem can be solved by PSO if a proper fitness function is defined. In PSO every particle finds a solution to the optimization problem. The position of a particle is changed according to the best position experienced by it and that of the best particle in the entire swarm. The closeness of the particle to the global optimum is measured using the predefined fitness function. Each particle in the group is defined in the following format:

- x_i the present position of the particle;
- v_i the present velocity of the particle;
- p_i the local best position of the particle.

Based on the above definitions, the position of a particle changes according to:

$$V_i(t+1) = i_w \cdot v_i(t) + r_1 \cdot a_{c1}(P_i(t) - X_i(t)) + r_2 \cdot a_{c2}(G_i(t) - X_i(t)) \tag{1}$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \tag{2}$$

Here, variables r_1 and r_2 are positive random numbers, produced by a uniform distribution and limited by an upper bond. Coefficients a_{c1} and a_{c2} are named acceleration constants and i_w is called the inertia weight. Moreover, p_i is the local best solution found up to now by the i-th particle, while g_i shows the position of the best particle thus far in the entire swarm. Equation (1) represents the new velocity of the particle expressed as the sum of the three objective functions explained below.

1. *Fraction of the old velocity* that depends on the inertia coefficient defined at the beginning of the algorithm and which reduces gradually.
2. *Perceptive factor,* which is the distance of the particle from its best visited position multiplied by an acceleration coefficient and a random value to train the particle not to fly far from its best experience.
3. *Group factor,* which is the distance of the particle from the global best visited position multiplied by an acceleration coefficient and a random value to train the particle to follow the best swarm to converge to the goal of the algorithm.

The local best position of particle i is the best position experienced by particle i up to now. If f is the fitness function then the personal best position of a particle at time step t is equal to:

$$P_i(t+1) = \begin{cases} P_i(t) & \text{if } f(X_i(t+1)) > f(P_i(t)) \\ X_i(t+1) & \text{if } f(X_i(t+1)) < f(P_i(t)) \end{cases} \tag{3}$$

After completing the algorithm, a large number of particles are predicted to converge within a small radius around the global optimum of the flying space.

4 PSO Data Clustering Algorithm

A clustering algorithm such as PSO-clustering can be implemented for automatic classification of the data of the output graph of real-time PCR. PSO has become a popular and powerful method in cluster analysis, and has been applied in many fields [14]. PSO is a data clustering technique based on the optimization of the objective function [10]:

$$J_e = \frac{\sum_{i=1}^{C} \sum_{j=1}^{N} \|x_j - y_i\|^2 / n_i}{C} \quad \forall x_j \in n_i \tag{4}$$

where C is the number of clusters and N the number of data. Each data point in the data set needs to belong to a cluster. The purpose of PSO is to group data points into different specific clusters. Let $X = \{x_1, x_2, \ldots, x_n\}$ be a collection of data. By minimizing (4), X is classified into C separate clusters, n_i is the number that shows whether data x_j is a member of the specific cluster and cluster set y_i, and $\mathrm{Y} = \{\mathrm{y}_1, y_2, \ldots, y_n\}$ includes all the cluster centers.

Here, $\|x_j - y_i\|$ is the Euclidean distance between x_j and y_i.

The aim of this study is to cluster the results of the TaqMan reactions, called “YES” and “NO” reactions. In addition, each graph of the reactions is depicted as a vector $X_j = \{x_{j(1)}, x_{j(2)}, \ldots, x_{j(45)}\}$. The reactions are clustered into two groups with their centroids at $Y_j = \{y_{1(1)}, y_{1(2)}, \ldots, y_{1(45)}\}$ and $Y_2 = \{y_{2(1)}, y_{2(2)}, \ldots, y_{2(45)}\}$, respectively. The center that exists in the non-amplification part is definitely smaller than the other center in the amplification part. Based on the previous description, we refer to these centers as the “YES” and “NO” centers, where the “NO” center is smaller than the “YES” center. Finally, by using the PSO data clustering algorithm these data are exploited to cluster the TaqMan reactions into “YES” and “NO” reactions.

The whole classification process for automatic classification of a datum of the output graph of real-time PCR by using PSO can be described by the following steps.

```
For p = 1 to 45
    Initialize each particle with 2 random cluster centers.
    For iteration iter = 1 to maximum iterations do
        Reduce inertia from 0.9 to 0.4 based on the iterations
        For all particles $i$ do
            For all members $x_j$ in the swarm do
                Calculate Euclidean distance of $x_j$ with all cluster centers
                Appoint $\mathrm{x}_j$ to the group that have nearest center to $\mathrm{x}_j$
            End for
            Compute the $J_e$
        End for
        Find the personal best and global best position of each particle.
        Update position and velocity formula of PSO (1, 2).
    End for
    Define the members of a cluster with the larger centroid as YES and the other as NO
End for
```

The initial parameters for PSO clustering are as follows:

- number of iterations is 90;
- total number of input data is 6;
- number of cluster centers is 2;
- number of particles is 30;
- inertia reduces from 0.9 to 0.4;
- correction factor = 1.42;
- random coefficient is a variable between 0 and 1.

Tables 1 and 2 contain data that can be clustered by this particular algorithm.

5 Results and Discussion

As mentioned previously, in the in vitro phase of the readout approach, each real-time PCR reaction is mapped to a binary output (either "YES" or "NO"), based on the occurrence or absence of an exponential amplification. Given the existence of this mapping, the subsequent in silico information processing is capable of determining the Hamiltonian path of the input instance (that is, $V_0 \rightarrow V_2 \rightarrow V_4 \rightarrow V_1 \rightarrow V_3 \rightarrow V_5$ in the example instance). PSO is used to cluster the TaqMan reactions produced by the DNA LightCycler. The parameters for clustering are $N = 6$ and $C = 2$, with the algorithms implemented using MATLAB 7.0. Figure 4 depicts the convergence curve of the PSO-clustering algorithm.

The allocation of data to the partitions is given in Tables 3 and 4. Based on these tables, it is shown that the PSO algorithm correctly clusters the "YES" and "NO" reactions compared to manual attempts. Moreover, the final result is similar to that of previous work by Saaid et al. [9], although in the Fuzzy method the values for M1j and M2j are floating point numbers between 0 and 1. Recently, the efficiency of the PSO approach has been widely discussed and a novel PSO-NTVE method has been proposed [15]. In this study, the software has been tested repeatedly and the robustness of the obtained result shown.

This software can analyze only six nodes of the HPP. As shown in Figs. 5 and 6, the output data can be plotted to illustrate the difference between the "YES" and "NO" reactions. According to Table 3, the partition matrix (grey color) consists of either 0's or 1's. Hence, if this matrix, M, is defined as

$$M = \begin{bmatrix} 01 \\ 10 \\ 01 \\ 10 \\ 10 \\ 01 \end{bmatrix}$$

Table 1 First set of input data for clustering algorithm

Cycles	TaqMan 1	TaqMan 2	TaqMan 3	TaqMan 4	TaqMan 5	TaqMan 6
1	−0.06153	−0.05134	−0.04466	−0.07269	−0.02776	−0.05634
2	−0.0485	−0.03884	−0.03059	−0.06071	−0.0267	−0.04297
3	−0.02918	−0.02599	−0.02849	−0.03151	−0.01737	−0.02627
4	−0.00942	−0.00632	−0.00209	−0.00712	−0.0066	−0.00402
5	0.02244	0.020956	0.017757	0.029153	0.013787	0.026531
6	0.064662	0.050194	0.043412	0.070199	0.036876	0.046726
7	0.111083	0.077998	0.074357	0.118585	0.06485	0.083573
8	0.159726	0.10396	0.107139	0.166418	0.09142	0.117665
9	0.214736	0.122577	0.131925	0.211994	0.113661	0.150973
10	0.26243	0.13438	0.147138	0.254498	0.122613	0.184741
11	0.308589	0.142117	0.156186	0.286811	0.136738	0.217243
12	0.348588	0.146311	0.159987	0.310461	0.141073	0.24109
13	0.381277	0.151949	0.163526	0.325078	0.143175	0.262244
14	0.404939	0.153523	0.164836	0.336238	0.148628	0.277632
15	0.418205	0.159166	0.164573	0.342774	0.155203	0.286288
16	0.43205	0.161658	0.163918	0.345082	0.155341	0.291667
17	0.438394	0.163752	0.165228	0.347964	0.160937	0.299558
18	0.444454	0.167033	0.165754	0.349508	0.161217	0.301093
19	0.451093	0.171226	0.165889	0.347775	0.162478	0.303976
20	0.450514	0.171361	0.162608	0.350272	0.166813	0.305323
21	0.465517	0.171997	0.165099	0.350083	0.16765	0.304173
22	0.464359	0.172957	0.165491	0.345854	0.163176	0.304362
23	0.473881	0.175454	0.167065	0.34835	0.169471	0.302826
24	0.481098	0.175454	0.165099	0.345854	0.173526	0.304173
25	0.488021	0.180455	0.166146	0.347964	0.176184	0.307055
26	0.490041	0.181802	0.166544	0.345082	0.179121	0.304362
27	0.496089	0.183345	0.170872	0.342003	0.183528	0.304173
28	0.490903	0.186039	0.164836	0.342963	0.18059	0.301668
29	0.492923	0.184692	0.163397	0.338734	0.184361	0.302629
30	0.492628	0.187763	0.162871	0.336041	0.186045	0.299944
31	0.496089	0.187763	0.161034	0.339506	0.18919	0.300322
32	0.496964	0.192	0.161298	0.338545	0.196948	0.302826
33	0.499551	0.194308	0.156578	0.341428	0.200522	0.300708
34	0.499846	0.196419	0.157759	0.337199	0.21017	0.302826
35	0.499551	0.201616	0.15697	0.344507	0.215626	0.303401
36	0.499267	0.204688	0.160906	0.340853	0.223177	0.304173
37	0.499551	0.206609	0.154741	0.341814	0.226322	0.303598
38	0.500708	0.211232	0.158672	0.342199	0.236812	0.306094
39	0.503307	0.216422	0.15828	0.33912	0.244158	0.303212
40	0.499551	0.220077	0.157759	0.341625	0.2536	0.302629
41	0.507347	0.225267	0.159724	0.340081	0.262621	0.304173
42	0.501866	0.228732	0.156578	0.336624	0.269966	0.304937
43	0.503886	0.234119	0.158543	0.33797	0.279194	0.302251
44	0.497826	0.240655	0.158151	0.336813	0.288636	0.302826
45	0.503886	0.24642	0.161163	0.334891	0.304787	0.301093

to obtain the elements of matrix M, in every iteration the Euclidean distance between each datum in the dataset and the position of all the particles in that specific iteration is computed and stored in a 6×2 matrix. Then the minimum element of every row of this matrix is identified and stored in a six element array. Thereafter, every element of every row of the aforementioned matrix is compared with the corresponding value from the specific array and if the element of the matrix is smaller, the corresponding value in matrix M is "1". Otherwise, it is "0". In matrix M, the first column gives the assignment of data into the "YES" cluster if the value is 1 for each particular TaqMan, whereas the second column shows the assignment of data into the "NO" cluster for each particular TaqMan. If there are more than two cluster centers, only one column can have the value 1 in each row, while all the other rows have 0's. The implemented software in this work has the ability to cluster the TaqMan reactions by applying the PSO algorithm using MATLAB. It can be seen that between steps 35 and 45 in the first data set and between steps 28 and 43 in the second data set, there is a change in the clustering trend. The main reason for this phenomenon is noise that exists in the environment in which the experiment was carried out. However, based on the specific rule that the average number of clusters justifies the final result and that the steps of the experiment are not independent, these sudden changes

Table 2 Second set of input data for clustering algorithm

Cycles	TaqMan 1	TaqMan 2	TaqMan 3	TaqMan 4	TaqMan 5	TaqMan 6
1	−0.03709	−0.06681	−0.04673	−0.05613	−0.06094	−0.02923
2	−0.02765	−0.05614	−0.03441	−0.04205	−0.04793	−0.02039
3	−0.01847	−0.03542	−0.02182	−0.02771	−0.02905	−0.01943
4	−0.00536	−0.00858	−0.00609	−0.00822	−0.00796	−0.0072
5	0.014311	0.027781	0.018393	0.021025	0.021148	0.012386
6	0.037172	0.072365	0.043919	0.056954	0.063791	0.034637
7	0.063263	0.13111	0.076046	0.092882	0.11241	0.062436
8	0.083851	0.19007	0.10056	0.13209	0.17394	0.081316
9	0.11611	0.24661	0.12102	0.17136	0.23987	0.098364
10	0.13119	0.29122	0.1382	0.21309	0.30525	0.10558
11	0.14142	0.31527	0.15525	0.25232	0.37133	0.11489
12	0.14574	0.32719	0.15551	0.28578	0.42067	0.11803
13	0.14679	0.33469	0.15983	0.31617	0.46481	0.11895
14	0.14915	0.34354	0.17059	0.33983	0.48452	0.11843
15	0.14561	0.34738	0.17833	0.35598	0.50689	0.11882
16	0.15033	0.34815	0.18187	0.37425	0.51493	0.12052
17	0.14902	0.34834	0.18816	0.38175	0.52262	0.12092
18	0.15125	0.34738	0.1933	0.39099	0.52858	0.11947
19	0.14915	0.34585	0.19637	0.39675	0.53802	0.12026
20	0.15138	0.34623	0.20137	0.40974	0.54571	0.12013
21	0.15309	0.34334	0.20541	0.40598	0.5541	0.12223
22	0.14627	0.33796	0.21426	0.41666	0.56074	0.12039
23	0.14719	0.342	0.22022	0.4181	0.56739	0.11961
24	0.146	0.34585	0.22484	0.43426	0.57158	0.11895
25	0.15728	0.34065	0.22984	0.4406	0.57507	0.12157
26	0.15099	0.34315	0.23542	0.44262	0.57997	0.12249
27	0.15217	0.34642	0.246	0.44955	0.58417	0.12236
28	0.14863	0.34661	0.24945	0.44204	0.58278	0.12079
29	0.15178	0.34507	0.2533	0.4406	0.58906	0.12105
30	0.14968	0.34373	0.25888	0.45069	0.58872	0.11803
31	0.15072	0.33546	0.26157	0.44406	0.59326	0.12052
32	0.14915	0.33623	0.27003	0.44925	0.59011	0.11987
33	0.14915	0.33007	0.27234	0.44291	0.59187	0.12249
34	0.1502	0.33507	0.27792	0.45387	0.59711	0.12039
35	0.1544	0.33142	0.28638	0.45329	0.59641	0.1221
36	0.14929	0.33258	0.29638	0.45617	0.60165	0.12236
37	0.14745	0.32911	0.29677	0.45907	0.59607	0.12079
38	0.15047	0.32854	0.30754	0.45069	0.59675	0.1242
39	0.14627	0.33103	0.31369	0.45387	0.60235	0.12275
40	0.14771	0.33469	0.32138	0.45877	0.60759	0.12367
41	0.14915	0.33911	0.33254	0.46051	0.6048	0.12406
42	0.15125	0.32815	0.3412	0.45791	0.60446	0.12551
43	0.15401	0.33373	0.34561	0.45993	0.60376	0.12354
44	0.15519	0.3345	0.36274	0.45963	0.61074	0.12302
45	0.15532	0.33238	0.36639	0.45271	0.61285	0.12485

are not considered in allocating data to a specific cluster. For example, in the first data set, for TaqMan 4 all data are allocated to the "YES" cluster until step 35. Although after step 35 and until step 45 the data are allocated to the "NO" cluster, based on the nearest Euclidean distance (as depicted in Figs. 5 and 6), by applying the rule explained above, all the steps in TaqMan 4 are allocated to the "YES" cluster in the first data set.

Table 5 shows the final results of the data clustering from the first to the last steps in both data sets.

6 Conclusions

The PSO data clustering algorithm has been implemented to automate the in silico information of the real-time PCR. This PCR based readout approach for DNA computing was implemented on the LightCycler System. Moreover, experimental results show that the amplification response for "YES" and "NO" reactions can be

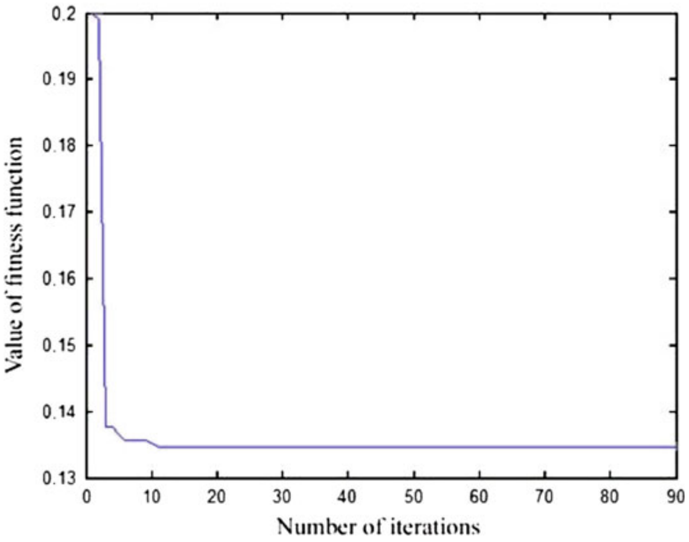

Fig. 4 Convergence curve of the fitness function

Table 3 PSO partition value for each TaqMan reaction in the first data set

TaqMan	M1j	M2j	Manual observation	Fuzzy C-means	PSO
TaqMan(v_0, v_1, v_2)	1	0	“YES”	“YES”	“YES”
TaqMan(v_0, v_1, v_3)	1	0	“YES”	“YES”	“YES”
TaqMan(v_0, v_1, v_4)	0	1	“NO”	“NO”	“NO”
TaqMan(v_0, v_2, v_3)	1	0	“YES”	“YES”	“YES”
TaqMan(v_0, v_2, v_4)	0	1	“NO”	“NO”	“NO”
TaqMan(v_0, v_3, v_4)	0	1	“NO”	“NO”	“NO”

Table 4 PSO partition value for each TaqMan reaction in the second data set

TaqMan	M1j	M2j	Manual observation	Fuzzy C-means	PSO
TaqMan(v_0, v_1, v_2)	0	1	“NO”	“NO”	“NO”
TaqMan(v_0, v_1, v_3)	1	0	“YES”	“YES”	“YES”
TaqMan(v_0, v_1, v_4)	0	1	“NO”	“NO”	“NO”
TaqMan(v_0, v_2, v_3)	1	0	“YES”	“YES”	“YES”
TaqMan(v_0, v_2, v_4)	1	0	“YES”	“YES”	“YES”
TaqMan(v_0, v_3, v_4)	0	1	“NO”	“NO”	“NO”

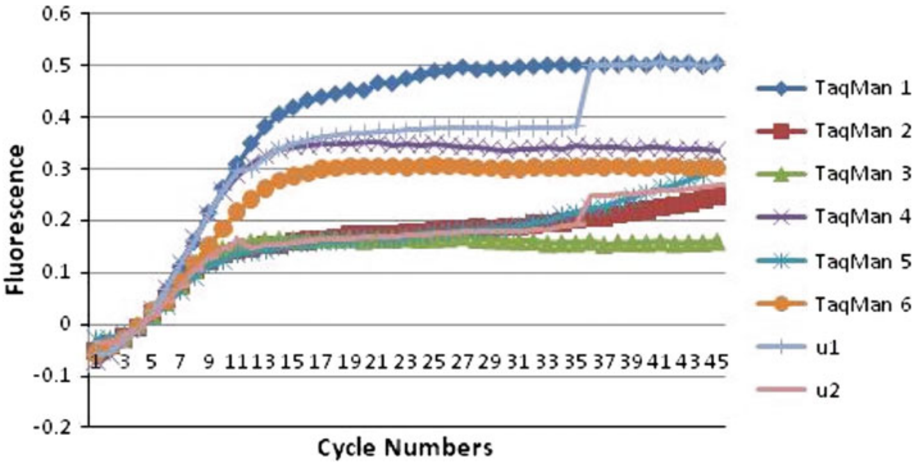

Fig. 5 Output of real-time PCR with “YES” and “NO” centers calculated using PSO clustering algorithm (*first data set*)

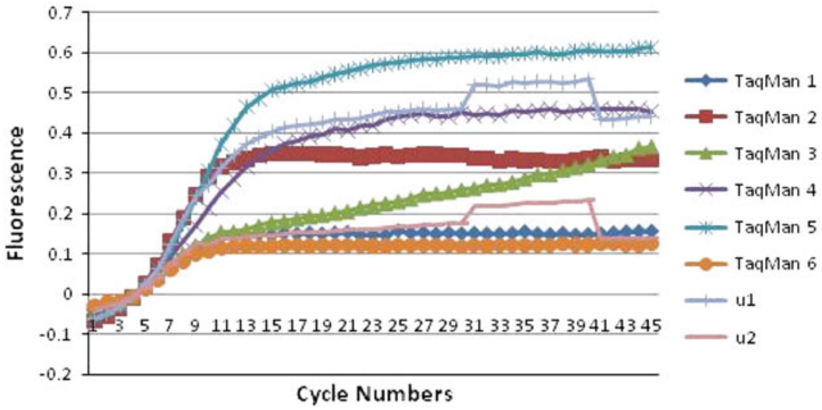

Fig. 6 Output of real-time PCR with "YES" and "NO" centers calculated using PSO clustering algorithm (*second data set*)

Table 5 Cluster centers for both data sets

u1	u2	u1	u2
−0.06129	−0.03768	−0.06047	−0.03621
−0.0487	−0.02748	−0.05461	−0.03477
−0.03073	−0.01991	−0.02829	−0.01737
−0.00799	−0.00572	−0.00737	−0.00306
0.023318	0.01503	0.027842	0.018735
0.064369	0.038576	0.067431	0.044302
0.121761	0.073657	0.114834	0.075195
0.182006	0.099454	0.163072	0.105046
0.24324	0.126719	0.213365	0.129784
0.269854	0.12499	0.258464	0.147218
0.312971	0.137187	0.2977	0.163071
0.344548	0.13976	0.300047	0.149124
0.371889	0.141857	0.322866	0.152884
0.389291	0.146057	0.339603	0.155662
0.403409	0.147588	0.349089	0.159648
0.412444	0.15091	0.356266	0.160305
0.417571	0.1527	0.361972	0.163306
0.422234	0.154665	0.365018	0.164668
0.426875	0.155261	0.367615	0.166531
0.433893	0.157625	0.368703	0.166927
0.434476	0.160245	0.373257	0.168248
0.43845	0.160298	0.371525	0.167208
0.442477	0.162339	0.375019	0.170663
0.450563	0.163264	0.377042	0.17136
0.452105	0.169563	0.381014	0.174262
0.455254	0.169637	0.379828	0.175822
0.460048	0.173507	0.380755	0.179248
0.457143	0.172955	0.378512	0.177155
0.458238	0.175376	0.378096	0.177483
0.461047	0.175533	0.376204	0.178893
0.518656	0.217068	0.378639	0.179329
0.519682	0.218818	0.379445	0.183415
0.51739	0.218509	0.380562	0.183802
0.525495	0.220896	0.379957	0.188116
0.524847	0.223576	0.382486	0.191404
0.528905	0.225152	0.499267	0.246759
0.527565	0.223529	0.499551	0.246617
0.523722	0.227687	0.500708	0.251002
0.528112	0.228438	0.503307	0.252238
0.533182	0.231848	0.499551	0.255138
0.434242	0.136604	0.507347	0.258373
0.432929	0.138373	0.501866	0.259367
0.435742	0.138775	0.503886	0.262415
0.441903	0.139105	0.497826	0.265416
0.441084	0.140074	0.503886	0.269671

categorized separately. It is possible that other suitable algorithms can be implemented to cluster the output of real-time PCR automatically.

References

1. Mullis, K.; Faloona, F.; Scharf, S.; Saiki, R.; Horn, G.; Erlich, H.: Specific enzymatic amplification of DNA in vitro: the polymerase chain reaction. Cold Spring Harb. Symp. Quant. Biol. **51**, 263–273 (1986)
2. Overbergh, L.; Giulietti, A.; Valckx, D.; Decallonne, B.; Bouillon, R.; Mathieu, C.: The use of real-time reverse transcriptase PCR for the quantification of cytokine gene expression. J. Biomol. Tech. **14**(1), 33–43 (2003)
3. Walker, N.J.: A technique whose time has come. Science **296**, 557–559 (2002)
4. Lakowicz, J.R.: Principles of Fluorescence Spectroscopy, vol. 3. Kluwer Academic/Plenum Publishers, New York (2006)
5. Ibrahim, Z.; Rose, J.A.; Tsuboi, Y.; Ono, O.; Khalid, M.: A new readout approach in DNA computing based on real-time PCR with TaqMan probes. In: Mao, C.; Yokomori, T. (eds.) Lecture Notes in Computer Science (LNCS), vol. 4287. Springer-Verlag, pp. 350–359 (2006)
6. Adleman, L.M.: Molecular computation of solutions to combinatorial problems. Science **266**, 1021–1024 (1994)
7. Rose, J.A.; Deaton, R.; Garzon, M.; Stevens, S.E. Jr.: The effect of uniform melting temperatures on the efficiency of DNA computing. In: DIMACS Workshop on DNA Based Computers III, pp. 35–42 (1997)
8. Wood, D.H.; Clelland, C.L.T.; Bancroft, C.: Universal biochip readout of directed Hamiltonian path problems. Lect. Notes Comput. Sci. **2568**, 168–181 (1999)
9. Saaid, M.F.M.; Ibrahim, Z.; Khalid, M.; Sarmin, N.H.; Rose, J.: Fuzzy C-means clustering for DNA computing readout method implemented on LightCycler system. In: Proceedings of SICE International Conference. Chofu, Japan, 20–23 August 2008, pp. 676–681 (2008)
10. Omran, M.; Salman, A.; Engelbrecht, A.P.: Image classification using particle swarm optimization. In: Conference on Simulated Evolution and Learning, vol. 1, pp. 370–374 (2002)
11. Kennedy, J.; Eberhart, R.C.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, Perth, Australia, vol. 4, pp. 1942–1948 (1995)
12. Kennedy, J.; Eberhart, R.C.; Shi, Y.: Swarm Intelligence. Morgan Kaufmann Publishers, San Francisco (2001)
13. Poli, R.: Analysis of the publications on the applications of particle swarm optimization. J. Artif. Evol. Appl. **2008**:685175 (2008)
14. Abraham, A.; Das, S.; Roy, S.: Swarm intelligence algorithms for data clustering. In: Maimon, O.; Rokach, L. (eds.) Soft Computing for Knowledge Discovery and Data Mining. Springer Verlag, Germany, pp. 279–313 (2007)
15. Ko, C.N.; Chang, Y.P.; Wu, C.J.: An orthogonal-array-based particle swarm optimizer with nonlinear time-varying evolution. Appl. Math. Comput. **191**, 272–279 (2007)